Security of Infocommunication Systems and Internet of Things

2026 Vol 4, No 1

https://doi.org/10.31861/sisiot2026.1.01010

Received 17 April 2026; revised 30 May 2026; accepted 15 June 2026; published 30 June 2026

An Ontology-driven Cloud Platform for Intelligent Monitoring with Dashboard-centric Analytics and Domain Knowledge Integration

Vasyl Lyashkevych*

System Design Department, Faculty of Electronics and Computer Technologies, Ivan Franko National University of Lviv, Lviv, Ukraine

*Corresponding author (E-mail: vasyl.liashkevych@lnu.edu.ua)

ABSTRACT This paper presents an ontology-driven cloud platform for intelligent monitoring that unifies domain knowledge representation, cloud-native telemetry acquisition, multi-method analytics, and dashboard-centric decision support in a single adaptive environment. Unlike conventional monitoring solutions that mainly aggregate metrics, logs, and alerts, the proposed platform treats the monitored information system as a dynamic, domain-dependent object whose state must be observed, diagnosed, predicted, and semantically interpreted with respect to context, architecture, and operational constraints. The platform integrates domain ontologies, telemetry pipelines, analytical and modeling services, hierarchical monitoring nodes, and an interactive dashboard layer supporting descriptive, diagnostic, predictive, prescriptive, comparative, and context-adaptive analytics. The study is motivated by the transition from fragmented metric-oriented monitoring to knowledge-driven monitoring capable of combining heterogeneous data sources, domain semantics, and adaptive decision support. The architecture is organized as a layered cloud platform, comprising data acquisition, integration and streaming, storage, ontology and knowledge management, analytical services, hierarchical federated monitoring, self-adaptation, and dashboard interaction. A central feature of the platform is the integration of domain knowledge into the monitoring loop through ontological mappings, semantic enrichment of telemetry, rule-based event interpretation, and dashboard widgets grounded in domain concepts. This allows the platform not only to detect anomalies and degradation patterns, but also to explain them in the language of the target domain and to adapt monitoring configuration to changes in the monitored system state. The paper also outlines thematic use cases showing how the same platform core can be connected to different application domains through ontology resources and domain-specific analytical templates. The proposed approach provides a reference basis for semantically integrated monitoring platforms.

KEYWORDS intelligent monitoring, dashboard analytics, domain knowledge integration, federated monitoring, self-adaptive monitoring.

I. INTRODUCTION

Modern information systems (ISs) operate in increasingly dynamic, heterogeneous, and distributed environments where software services, cloud resources, user interactions, external APIs, contextual factors, and domain events collectively influence the operational behavior and state of the IS. In such environments, traditional monitoring approaches, based primarily on isolated metrics, static thresholds, or individual log streams, are no longer sufficient for informed decision-making. Recent research in software analytics and software development analytics shows a clear shift from narrow metric observation to integrated analytical environments that combine multiple development and operational artifacts to support decisions [1, 2]. At the same time, state-oriented interpretation of software and ISs is becoming increasingly important, since the monitored object should be understood not as a set of independent signals, but as a dynamic system whose state changes under the influence of architecture, workload, context, and risks [3].

The problem becomes even more pronounced in cloud-native and service-oriented environments. Cloud applications are typically elastic, distributed, and structurally variable, while monitoring pipelines must process large volumes of metrics, logs, traces, and events across multiple infrastructure and application layers. Research on cloud-native architectures and autonomous management shows that monitoring in such environments must be scalable, adaptive, and tightly coupled to the service structure and execution context [4, 5]. However, even with modern observability mechanisms in place, many monitoring solutions remain predominantly telemetry-driven. They can effectively collect and visualize technical signals, but they often lack a consistent mechanism for transforming these signals into domain-relevant knowledge about the monitored IS.

Another important challenge is the analytical interpretation of data at runtime. Current research shows that anomaly detection, especially in logs and time series, is highly dependent on the quality of data preparation, parsing strategy, contextualization, and integration into operational workflows [6, 7]. This means that intelligent monitoring (IM) cannot be reduced to the application of a single machine learning model or dashboard widget. Instead, it requires a structured analytical process in which heterogeneous data are interpreted with respect to domain semantics, monitored states, and management objectives.

This creates the need for a stronger knowledge integration layer.

Recent research on semantic integration, context retrieval, and meaning-driven analytics shows that domain models, contextual representations, and semantically enriched analytics pipelines can significantly improve the interpretability and usefulness of monitoring results [8-10]. In parallel, ontological approaches are increasingly being used to support robust autonomy, enterprise integration, semantic interoperability, and knowledge-based coordination [11, 12]. These results suggest that ontologies are not simply supporting documentation artifacts; rather, they can function as operational resources that link monitored objects, events, states, dependencies, risks, and analytical rules within an IM platform.

The visualization layer also needs to be revisited. In many practical systems, dashboards remain the final interfaces for displaying indicators, trends, and alarms. However, recent work on knowledge-based visualization, ontology-supported observability, and strategic dashboard construction shows that dashboard environments can be designed as active decision-support surfaces that integrate metrics, semantic explanations, incident context, and response instructions [13-16]. This is especially important for IM, where the dashboard should not only show what happened, but also help interpret why it happened, what it means in the target domain, and how the monitoring or management process should respond.

A further dimension of the problem concerns self-adaptation and distributed monitoring. Studies on cloud-native recovery, hierarchical federated learning, and self-adaptive federated systems show that large-scale monitoring environments increasingly require local autonomy, distributed learning, adaptive coordination, and multi-layer aggregation of monitoring knowledge [17-20]. At the same time, scenario-based monitoring applications and domain-based platforms confirm the practical importance of flexible architectural patterns that can be reused across environmental, operational, cyber-physical, and service-oriented domains [21-25]. Ukrainian research by S. V. Holub et al. further demonstrates the relevance of multi-agent monitoring ISs, multilayer monitoring agents, and IM models for complex distributed environments [26-30]. Additional Ukrainian work with ontological intelligent agents and ontological ISs further confirms the importance of explicit semantic knowledge as a basis for coordinated monitoring and interpretation [31, 32].

Therefore, the key scientific and practical problem can be formulated as follows: how to develop an ontology-driven cloud platform for IM that not only collects heterogeneous telemetry, but also integrates domain knowledge, supports multi-method analytics and presents results through a dashboard-centric analytical interface, and can be extended to self-adaptive hierarchical federated monitoring.

This paper proposes an ontology-driven cloud platform for IM with dashboard-centric analytics and domain knowledge integration. The central idea is to combine telemetry collection, semantic modeling, analytical services, and dashboard-based decision support within a single cloud architecture, where ontologies act as integration resources that link data sources, monitored objects, system states, events, risks, contextual factors, and analytical procedures. In this formulation, monitoring is seen as a knowledge-driven process of observation, diagnosis, prediction, explanation, and response support, rather than as a purely technical function of aggregating metrics.

The aim of the research is to develop the architectural foundations of an ontology-driven cloud platform for IM that supports domain knowledge integration, dashboard-centric analytics, and further extension to self-adaptive hierarchical federated monitoring. To achieve this goal, the article focuses on:

- identifying the limitations of current monitoring approaches;
- systematizing the role of semantic and ontological resources in IM;
- proposing a multi-tiered platform architecture;
- defining the role of the dashboard as an analytical and decision support environment;
- outlining how the platform can function as a diagnostic, adaptive monitoring loop over a monitored IS.

The scientific novelty of the research is that:

- an ontology-driven cloud platform for IM is proposed as a unified environment integrating telemetry pipelines, semantic knowledge resources, analytical services, dashboard-centric decision support, and adaptive monitoring management, and reconfiguration of monitoring logic under drift and evolution of the monitored object;
- a domain knowledge integration mechanism is proposed by using ontological resources act as operational means for semantic mapping, formalization of monitored states, analytical binding, and synthesis of monitoring configurations;
- the dashboard-centric IM model is systematized, whereby dashboards are interpreted as analytical and decision-support workspaces, rather than solely as visualization tools;
- the framework of self-adaptive hierarchical federated monitoring is extended, enabling coordination of distributed models, allowing the platform to support coordination of distributed models, multi-layer state assessment, adaptive reconfiguration of the monitoring process itself, and shifting of monitoring configurations in response to structural, behavioral, and semantic evolution of the monitored system.

The remainder of the paper is organized as follows. Section II reviews related work, Section III formulates the research gap and objectives, Sections IV–VIII present the architecture and its extensions, Section IX discusses worked examples and validation logic, Section X illustrates the interface prototype, and Sections XI–XII provide discussion and conclusions.

II. RELATED WORK

The literature relevant to ontology-driven cloud platforms for IM can be structured into six interrelated thematic areas:

- software and system analytics;
- cloud-native monitoring and observability;

- runtime anomaly detection and operational intelligence;
- semantic integration, context, and ontology-driven monitoring;
- dashboard-centric analytical environments;
- self-adaptive, hierarchical, and distributed monitoring.

This thematic structuring is important because the proposed platform lies at the intersection of these areas, rather than in just one of them.

A. Software and system analytics. Software analytics and software development analytics provide a broad foundation for IM. Recent reviews show that software analytics has evolved from isolated repository mining and metrics-based assessment to integrated decision-support ecosystems combining development, operations, and organizational data [1, 2]. This shift matters because monitoring can no longer be treated only as the collection of technical signals. Instead, it should be embedded in broader analytical workflows linking data interpretation to engineering and management decisions. In this context, state-oriented interpretation of monitored systems becomes especially relevant. The concept of functional software state proposed in [3] reinforces this direction by emphasizing that monitored systems should be modeled through states and transitions rather than unrelated indicators.

At the same time, the software analytics literature still focuses mainly on artifacts and engineering workflows, while this research calls for a stronger integration of runtime telemetry, domain semantics, and adaptive platform behavior. Thus, software analytics provides an important methodological foundation, but is not yet a complete architectural answer to ontology-driven IM.

B. Cloud-native monitoring and observability. The second main focus is on cloud-native architectures, autonomous management, and observability. Research in this area shows that cloud environments require monitoring mechanisms that can handle elasticity, distribution, infrastructure variability, and service layer dependencies [4, 5]. These works emphasize scalability, multi-layer service representation, and adaptive management, which are essential for any modern monitoring platform deployed in cloud environments. More recent studies also emphasize observability-aware remediation and coordinated cloud-native response mechanisms [17].

However, the focus of this stream remains on the orchestration of infrastructure and services, rather than on the semantic interpretation of the monitored domain. In most cases, observability is considered as a technical property of cloud systems, rather than as a knowledge-based process of diagnosis, explanation, and contextual decision support. Consequently, the cloud monitoring literature provides architectural and operational principles, but does not sufficiently address the integration of domain knowledge.

C. Runtime anomaly detection and operational intelligence. The third line of research concerns runtime analytics, especially anomaly detection in logs and time series. These studies show that the quality of monitoring results depends not only on the training model itself, but also on data preparation, log parsing, context sensitivity, and integration of analytical results into operational workflows [6, 7]. This line of work is highly relevant because IM platforms must process runtime signals in a way that remains interpretable and practical.

However, most anomaly detection research remains focused on model performance, feature extraction, and operational deployment. They rarely provide a broader semantic integration layer that can relate anomaly evidence to domain concepts, states, organizational meaning, or adaptive strategy choices. Thus, runtime anomaly detection research offers powerful analytical tools, but does not by itself solve the problem of semantically informed IM.

D. Semantic integration, context, and ontology-driven monitoring. A fourth and particularly important area encompasses semantic integration, contextualization, ontologies, and knowledge-based coordination. Ukrainian work by Rohatskyi and Lyashkevych [8], Yakubovych and Lyashkevych [9], and Lyashkevych and Shuvar [10] demonstrate the growing importance of semantic integration, context capture, and meaning-based analytics in intelligent distributed systems. These works are particularly useful for this article, as they show that data interpretation is significantly improved when analytical processes are coupled with contextual models and explicit semantic structures.

A broader framework for this direction is provided by ontology-driven research in the areas of dependable autonomy, enterprise integration, and semantic coordination [11, 12]. These works demonstrate that ontologies can support more than just modeling: they can also act as operational resources for integration, interpretation, and coordination between heterogeneous system components. Additional work on knowledge-based representations of cloud patterns and services [13], ontology-driven visualization [14], and ontology-driven monitoring in serverless observability [15] further strengthens the argument that semantic knowledge can be directly embedded in monitoring and visualization environments.

This direction is also supported by other Ukrainian researchers. Rogushyna [31] shows the importance of ontological knowledge for intelligent agents, while Nesterenko [32] demonstrates the applicability of ontology-driven ISs in administrative environments. Together, these works support the idea that ontological resources can be used not only for static domain description, but also for guiding interpretation and interaction in monitoring systems.

Despite these achievements, most ontology-driven research focuses either on semantic modeling without a full-fledged cloud monitoring architecture, or on domain-specific implementations without a generalized monitoring platform perspective. Therefore, the literature still lacks a sufficiently unified model of an ontology-driven platform.

E. Dashboard-centric analytical environments. The fifth research area concerns dashboards, visual analytics, and knowledge-based visualization. Recent work has begun to view dashboards not simply as interfaces for displaying metrics, but as analytical and strategic environments that

structure user interactions with data, events, and decisions [14-16]. This is particularly relevant for IM, as the dashboard can serve as a central layer where analytical results, semantic explanations, and action recommendations for different user roles are integrated.

The role of semantically grounded dashboards becomes even clearer in works that link visualization to ontologically supported observability and knowledge-based service representations [13-15]. Such studies indicate that visualization should not be separated from the knowledge layer, as interpretation, navigation, and explanation depend on how the controlled entities and their relationships are formally represented.

However, the current literature on dashboards only partially addresses the specific needs of IM platforms, especially those that require state-oriented diagnostics, domain adaptation, and coordination between distributed components. Thus, dashboard research contributes analytical interaction, visual control, and decision-support structuring.

F. Self-adaptive, hierarchical, and distributed monitoring.

The sixth stream includes self-adaptive monitoring, federated learning, hierarchical aggregation, and agent-based distributed monitoring. Recent studies on hierarchical federated learning and self-adaptive federated systems show that distributed analytics environments increasingly rely on local autonomy, multi-layer aggregation, privacy-aware knowledge sharing, and adaptive coordination [18-20]. These ideas are highly relevant for large-scale IM, especially when the platform must work across heterogeneous subsystems, edge nodes, or organizationally distributed data sources.

This direction is complemented by scenario-oriented and subject-oriented platform studies [21-25], which show the practical importance of adaptive architectures that can be configured for different monitoring contexts. Ukrainian studies by Golub et al. provide additional support in terms of multi-agent monitoring ISs, multilayer monitoring agents, and machine learning-supported monitoring architectures [26-30]. These works are particularly important for this article, as they demonstrate a clear movement towards multilayer, agent-oriented, and adaptive monitoring structures in distributed environments.

However, the existing literature still addresses these topics in a fragmented manner. Federated learning research focuses on the coordination of distributed models. Agent monitoring research focuses on architectural or algorithmic organization. Cloud observability research focuses on operational signals. Ontology research focuses on semantics and dashboard research focuses on interaction. A unified view that integrates these components into a single ontology-driven cloud monitoring platform remains underdeveloped.

III. RESEARCH GAP AND OBJECTIVES

The literature review shows that substantial research already exists in several related fields, including software analytics, cloud-native monitoring, runtime anomaly detection, ontology-driven integration, dashboard-based analytics, and self-adaptive distributed monitoring [1-32]. However, these contributions remain largely fragmented and do not yet form a sufficiently unified and architectural foundation for designing an ontology-driven cloud platform for IM.

First, software analytics and software development analytics provide rich methods for extracting and interpreting information from development and operational artifacts, but they are not primarily aimed at creating a domain-aware cloud monitoring platform in which runtime telemetry, contextual knowledge, and architectural adaptation are integrated into a single operational environment [1-3]. Their main contributions are in the analytical perspective and decision support orientation, but they do not fully address semantic platform integration.

Second, studies on cloud-based monitoring and observability propose important principles of scalability, service-layer decomposition, and adaptive infrastructure management, but they mostly consider monitoring as a technical capability of cloud systems rather than as a domain-layer, knowledge-based diagnostic and explanation process [4, 5, 17]. This means that the infrastructure dimension is well represented in the literature, while the semantic and domain-related dimension is still not well formalized.

Third, research on runtime anomaly detection demonstrates significant progress in log analytics, time series intelligence, and operational diagnostics, but they typically focus on analytical pipelines, detection performance, and workflow integration rather than the broader problem of linking anomaly evidence to domain entities, semantic states, decision rules, and adaptive monitoring policies [6, 7]. Thus, they address an important subproblem of IM but not the full platform problem.

Fourth, ontology and context-oriented research shows that semantic integration, contextual modeling, and ontological reasoning can significantly improve interpretability, interoperability, and coordinated decision-making [8-15, 31, 32]. However, these studies often remain either domain-oriented, methodology-oriented, or without a generalized platform architecture. In other words, they demonstrate the value of semantics but do not yet provide a sufficiently general architecture for a cloud platform that combines ontologies with telemetry pipelines, analytics services, and dashboard-centric interactions.

Fifth, research on dashboards and visual analytics increasingly recognizes that dashboards can act not only as presentation tools but also as analytical and decision support environments [13-16]. However, dashboard-centric research typically focuses on presentation, visual interaction, or observability support and does not fully consider the integration of domain ontologies, adaptive analytics services, and hierarchical monitoring controls.

Finally, studies on federated, hierarchical, agent-based, and self-adaptive monitoring confirm the importance of local autonomy, multi-layer coordination, adaptive learning, and multilayer organization in distributed environments [18-30]. However, these contributions are still scattered among different research communities. Federated learning research focuses on the coordination of distributed models, agent-based monitoring research emphasizes organizational structure, and cloud observability research focuses on telemetry and

responsiveness. The concept of a single platform that unifies these ideas by integrating domain knowledge remains underdeveloped.

This gap is of scientific relevance, as IM increasingly requires a joint consideration of architecture, context, semantics, analytics, and adaptive response, rather than considering these dimensions in isolation. It is also of practical significance, as modern ISs operate in heterogeneous, dynamic, and domain-sensitive environments in which purely metric-oriented or infrastructure-oriented monitoring no longer provides sufficient support for informed management decisions.

The aim of the study is to develop the and architectural foundations of such a platform and to show how it can support domain-aware IM in complex information environments. To achieve this aim, the paper solves the following research tasks:

- to systematize the scientific background of IM at the intersection of software analytics, cloud observability, semantic integration, dashboard-based analytics, and adaptive distributed monitoring;
- to identify the limitations of existing approaches from the viewpoint of domain-aware cloud monitoring;
- to develop a layered reference architecture of an ontology-driven cloud platform for IM;
- to define a model of domain knowledge integration based on ontological resources;
- to substantiate the role of dashboard-centric analytics as an operational and decision-support layer of the platform;
- to outline the extension of the platform toward self-adaptive hierarchical federated monitoring.

Therefore, the scientific contribution of the article lies in the synthesis of several previously fragmented research directions into a single platform-oriented framework. Unlike existing works that usually emphasize either analytics, or monitoring infrastructure, or semantic modeling, or dashboards, or distributed adaptation, the proposed research integrates these directions into a holistic and architectural model. This allows us to consider IM not as an auxiliary operational subsystem, but as a knowledge-based platform for observation, diagnostics, prediction, explanation, and adaptive decision support in ISs.

IV. ARCHITECTURE OF THE ONTOLOGY-DRIVEN CLOUD PLATFORM

The proposed platform is designed as a layered cloud architecture supporting data collection, semantic integration, analytical processing, self-adaptation, and dashboard-based interaction. Fig. 1 presents the overall container-layer organization of the proposed ontology-driven cloud platform. It shows how the dashboard interface, API gateway, telemetry ingestion, analytics services, ontology and knowledge components, self-adaptation controller, and hierarchical federated coordinator interact in a single cloud environment. The diagram emphasizes that IM is not implemented as an isolated analytics tool, but as an integrated platform that combines telemetry processing, semantic interpretation, adaptive management, and user-centric decision support. The platform architecture includes the following main layers.

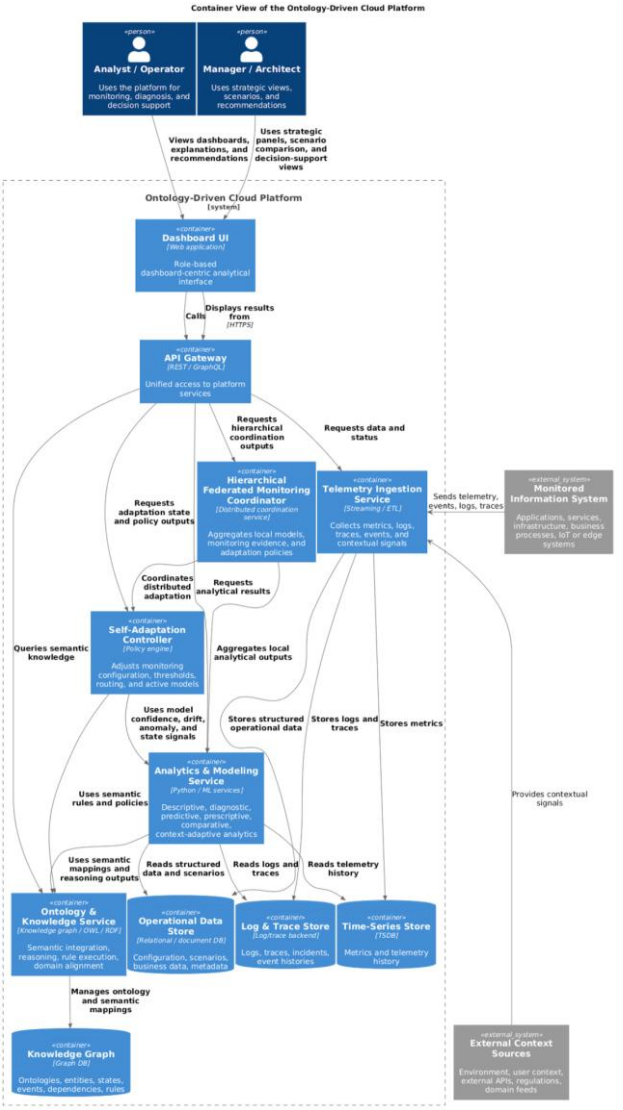

FIG. 1. C4 container architecture of the ontology-driven cloud platform.

A. Data acquisition layer. This layer is responsible for collecting domain-related telemetry and observations from disparate sources. These sources can include application metrics, event streams, execution logs, traces, business transactions, user interactions, security alerts, infrastructure metrics, configuration changes, and external contextual signals. Data can be obtained from ISs, cloud services, software repositories, IoT devices, enterprise applications, and domain-specific platforms.

The key challenge at this layer is not only the heterogeneity of formats, but also the heterogeneity of temporal scales, semantics, trust layers, and update rates. Therefore, the data collection layer must support streaming and batch data transfer, structured and unstructured representations, and integration with collectors and connectors.

B. Integration and streaming layer. This layer provides mechanisms for data ingestion, normalization, synchronization, routing, and event propagation. It may include message brokers, event buses, stream processors, data converters, ingestion APIs, and mediation services. The goal of this layer is to provide a unified platform-layer

model of events and observations suitable for further storage and analysis. The layer also applies metadata enrichment, including timestamps, source identifiers, monitored object identifiers, environment attributes, and initial domain bindings.

C. Hybrid storage layer. The platform requires multiple storage models because IM combines multiple categories of data:

- time series storage for metrics and signals;
- log stores for runtime records;
- trace storage for distributed interactions;
- document or relational storage for structured business and configuration data;
- graph-based storage for ontological resources and knowledge relationships.

This hybrid storage strategy allows the platform to preserve the unique analytical value of each data type while maintaining cross-layer interoperability through shared identifiers and semantic mappings.

D. Ontology and knowledge management layer. The ontology layer is the semantic core of the platform. It represents domain entities, monitored components, operational states, events, risks, dependencies, roles, contexts, policies, thresholds, actions, and analytical objects. Ontology resources enable semantic alignment of heterogeneous data, formalization of domain semantics, and construction of interpretable relationships between low-layer observations and high-layer domain concepts.

The ontology layer supports:

- modeling of domain concepts;
- mapping between telemetry and domain entities;
- taxonomies of states and events;
- interpretation of conditions and transitions based on rules;
- knowledge graph queries;
- semantic support for dashboard widgets and analytical services.

The presence of this layer allows us to move from monitoring at the signal layer to domain-aware monitoring.

E. Analytical and modeling services layer. This layer implements the platform's analytical methods and models. To comprehensively support IM, the platform includes several analytical modes:

- *descriptive analytics* to present the current state and analyze trends;
- *diagnostic analytics* to investigate the causes and explain anomalies;
- *predictive analytics* to predict degradation, incidents, or state transitions;
- *prescriptive analytics* to generate recommended actions or interventions;
- *comparative analytics* to evaluate alternatives, scenarios, or trajectories;
- *context-adaptive analytics* to adjust models and interpretations according to situational changes.

Services can include statistical methods, machine learning models, anomaly detectors, forecasting pipelines, rule-based reasoning, optimization modules, and modeling engines. Their work is based on both telemetry and ontology resources.

F. Hierarchical federated monitoring layer. For distributed ISs, the platform can be extended with a hierarchical federated monitoring layer. At lower layers, local monitoring nodes observe subsystems, train local models, and detect local anomalies or state changes. At intermediate layers, subsystem coordinators aggregate the output of local models and contextual information. At higher layers, the platform integrates cross-system knowledge, assesses the global state of the system, and coordinates monitoring strategies. This layer supports scalability, local autonomy, privacy preservation, and adaptive knowledge aggregation in multi-component environments.

G. Self-adaptation layer. Self-adaptation is a defining feature of IM in dynamic environments. Based on detected changes in monitoring state, drift patterns, reliability degradation, topology evolution, or contextual shifts, the platform can reconfigure parts of its own monitoring process. Adaptive actions can include:

- changing the sampling rate;
- adjusting alert thresholds;
- switching analytic models;
- updating semantic rules;
- enabling or disabling monitoring channels;
- rebalancing dashboard priorities;
- changing federation and aggregation policies.

In this way, the monitoring platform becomes not just an observer, but an adaptive diagnostic system.

H. Dashboard and interaction layer. The dashboard is the operational interface of the platform. It combines real-time metrics, trends, semantic annotations, alerts, state summaries, confidence indicators, forecasts, and recommendations in an integrated visual environment. The dashboard is role- and domain-specific: different views can be provided to analysts, operators, architects, managers, security engineers, and domain experts. It supports:

- KPI and state dashboards;
- anomaly and risk heat maps;
- trend and forecast charts;
- semantic explanations;
- scenario comparison panels;
- recommendation widgets;
- drill-downs into logs, traces, etc.

Thus, the dashboard is not a passive reporting tool, but a central interface for IM.

V. DOMAIN KNOWLEDGE INTEGRATION THROUGH ONTOLOGICAL RESOURCES

A fundamental feature of the platform (Fig. 2) is the ability to combine different subject domains without reworking the platform core. This is achieved by using ontological resources as domain integration mechanisms. Fig. 2 illustrates the process of integrating domain knowledge into a monitoring platform. It shows how domain documents, telemetry streams, contextual data, and expert knowledge are transformed into semantic representations through concept extraction, ontology modeling, rule definition, and knowledge graph construction. The diagram shows that domain integration is performed systematically through semantic mapping and analytical binding, which enables domain-aware notifications, explanations, and recommendations.

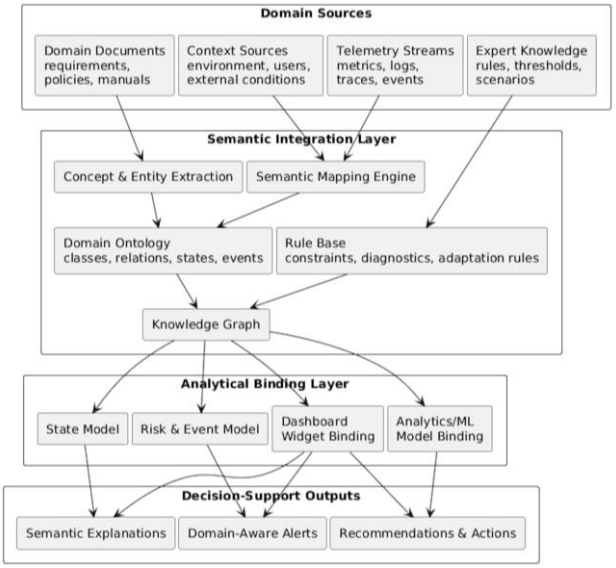

FIG. 2. Ontology-driven domain knowledge integration pipeline.

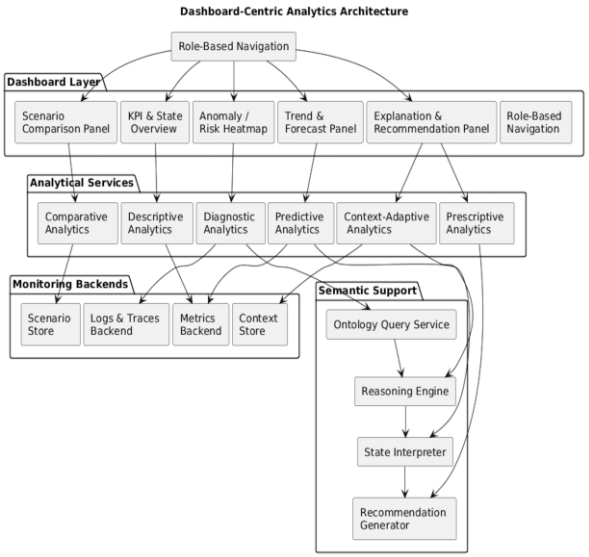

FIG. 3. Dashboard-centric analytics architecture.

The domain integration process includes these steps.

Step 1. Defining the domain scope. The target subject domain is analyzed in terms of monitored entities, states, events, goals, risks, roles, and decisions.

Step 2. Building or reusing an ontology. Domain concepts are formalized in an ontology using classes, relationships, attributes, constraints, and rules.

Step 3. Semantic mapping. Data sources and telemetry streams map to ontological entities and domain concepts.

Step 4. State formalization. Domain-specific states, deviations, risks, and transitions are represented formally.

Step 5. Analytical binding. Analytical methods are associated with specific entities, events, or state classes.

Step 6. Dashboard grounding. Dashboard views and widgets are associated with ontological concepts and semantic contexts.

This mechanism allows a single cloud platform to support multiple domains, such as software-intensive systems, cyber-physical systems, smart workspaces, service ecosystems, or enterprise operating environments [8, 9, 12, 20-24]. The reusable core of the platform remains stable, while the domain-specific layer is implemented through ontology resources and analytic configurations.

From a methodological perspective, ontologies play at least four roles in the platform:

- integration role, by connecting heterogeneous sources and models;
- interpretation role, by assigning semantic meaning to telemetry and events;
- reasoning role, by enabling rule-based inference and contextual explanations;
- adaptation role, by informing model selection and monitoring policies based on domain semantics.

This makes domain integration systematic, not ad hoc.

VI. DASHBOARD-CENTRIC ANALYTICS MODEL

The dashboard-centric model is the main contribution of the proposed platform (Fig. 3). Instead of viewing dashboards as end-point reporting interfaces, this approach views them as dynamic analytical workspaces where observation, diagnosis, prediction, explanation, and decision support are integrated.

Fig. 3 shows the internal analytical structure of a dashboard-centric monitoring model. It explains how the different components of a dashboard are connected to descriptive, diagnostic, predictive, prescriptive, comparative, and contextually adaptive analytical services, as well as semantic support mechanisms. The diagram confirms that the dashboard is not limited to visualization functions, but acts as an operational analytical layer, combining interpretation, comparison, explanation, and decision support.

A. Descriptive dashboard functions. Descriptive widgets display the current state of the monitored system through summary cards, metrics dashboards, trend graphs, topology maps, and aggregated status indicators. Their role is to provide a holistic operational picture of the monitored system and its domain context.

B. Diagnostic dashboard functions. Diagnostic components visualize anomalies, event correlations, state transitions, dependency paths, causal hypotheses, and semantic explanations. They support the investigation of the causes of deviation and its propagation in the monitoring environment.

C. Predictive dashboard functions. Predictive elements display forecasts, early warning indicators, degradation trajectories, expected state transitions, and uncertainty ranges. These features move the platform from reactive to proactive monitoring.

D. Prescriptive dashboard functions. Prescriptive widgets provide actionable recommendations, prioritized interventions, escalation options, or optimization suggestions. They can be generated using rules, optimization models, simulations, or hybrid reasoning pipelines.

E. Comparative and strategic dashboard functions. Comparative analytics allows the evaluation of scenarios, releases, subsystems, architectural alternatives, operating modes, or policy choices. Strategic dashboards support long-term management decisions, not just quick responses.

F. Context-adaptive dashboard functions. Context-aware widgets change their emphasis based on the current domain

context, system state, user role, or operating mode. For example, in high-risk situations, a dashboard might prioritize diagnostics and intervention recommendations, while in stable conditions, it might emphasize trends and optimization opportunities.

Thus, a dashboard-centric approach combines analytical depth with practical usability.

VII. CLOUD IMPLEMENTATION PRINCIPLES AND USE CASES

The proposed platform can be implemented as a cloud-native microservice architecture (Fig. 4). A practical implementation can include telemetry collectors, streaming infrastructure, scalable data stores, ontology and knowledge graph services, model serving components, API gateways, authentication modules, dashboard services, and orchestration mechanisms.

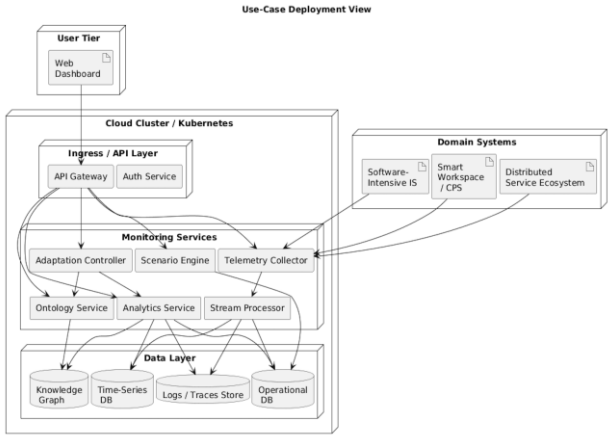

FIG. 4. Cloud deployment view of the monitoring platform.

Fig. 4 presents a deployment-oriented view of the proposed platform in a cloud environment. It shows the interaction between the user layer, the orchestration and cloud services layer, data storage components, and external monitoring systems from different application domains. The diagram demonstrates the practical feasibility of implementing the platform as a cloud-native microservice solution that can support multiple monitoring scenarios while maintaining a common architectural core.

A typical implementation can rely on the following functional cloud resources:

- containerized microservices;
- Kubernetes-based orchestration;
- telemetry collection and export services;
- message brokers and event streams;
- time series, document, and graph storage;
- analytical model services;
- dashboard and alerting services;
- identity and access control modules.

The architecture is suitable for thematic use cases in which monitoring requires both telemetry and semantic interpretation.

A. Use case: monitoring of software-intensive ISs. In this scenario, the platform tracks application services, deployment pipelines, incidents, technical debt indicators, security events, and operational telemetry. Ontologies formalize system components, software states, dependencies, risks, and intervention rules. The platform supports development, DevOps, quality assurance, security, and architecture decision-making.

B. Use case: monitoring of smart operational environments. In cyber-physical or intelligent work environments, the platform integrates sensor data, contextual signals, equipment status, environmental parameters, and human activity indicators. Domain ontologies describe monitoring areas, equipment classes, contextual states, alerts, and resource constraints. The platform supports context-aware risk monitoring and sustainability-oriented solutions.

C. Use case: monitoring of distributed service ecosystems. For enterprise platforms or service networks, a monitoring platform observes service-layer behavior, user interactions, SLA-related metrics, inter-service dependencies, and workload changes. Semantic models associate technical events with contractual, business, or organizational meaning.

These use cases demonstrate that the platform is domain-pluggable, not domain-specific.

VIII. TOWARD SELF-ADAPTIVE HIERARCHICAL FEDERATED IM

A particularly promising extension of the proposed architecture is self-adaptive hierarchical federated monitoring (Fig. 5).

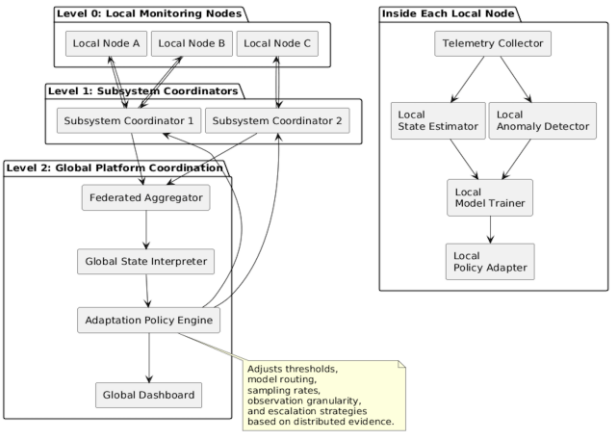

FIG. 5. Self-adaptive hierarchical federated monitoring architecture.

In complex distributed ISs, a fully centralized monitoring strategy may be inefficient, expensive, semantically weak, or impractical due to locality and data heterogeneity. A hierarchical federated approach provides a more suitable architectural model. Fig. 5 shows a hierarchical federated extension of the distributed IM platform. It shows how local monitoring nodes, subsystem coordinators, global aggregation services, and adaptation policies are organized across multiple layers of monitoring management. The diagram highlights that the proposed platform can evolve from centralized dashboard-centric analytics to distributed, adaptive, and federated monitoring management capable of self-tuning according to changes in system state.

At the local layer, individual nodes or subsystems monitor their own telemetry, detect anomalies, classify states, and update local models. At the aggregation layer, federated services integrate local knowledge, reconcile

interpretations, and derive monitoring conclusions at the regional or subsystem layer. At the global layer, the platform forms an integrated view of the monitored IS and issues adaptation policies.

Thus, the platform acts as a diagnostic and monitoring instrument over the IS itself. In this setting, the platform continuously addresses five key questions:

- What is the current state of the monitored system?
- Why did it reach this state?
- What is likely to happen next?
- How should the monitoring process itself adapt in response?
- Which monitoring components should be reconfigured, and at what layer?

The latter two questions are critical. Self-adaptation turns monitoring into a reflexive process in which the monitoring platform changes its own behavior in response to observed system evolution. This may involve retraining models, changing rule sets, modifying observation granularity, or reviewing dashboard priorities. Thus, self-adaptive hierarchical federated monitoring extends the capabilities of the platform from intelligent visualization and analytics to autonomous state-aware monitoring management.

IX. WORKED EXAMPLES AND VALIDATION STRATEGY

The proposed ontology-driven cloud platform is designed to monitor situations where conventional tools are insufficient. While traditional monitoring environments can collect metrics, logs, traces, and alerts, they remain limited when monitoring requires the joint integration of heterogeneous telemetry, domain semantics, state-oriented interpretation, dashboard-based explanation, and adaptive reconfiguration. Therefore, the validation in this study is and architecture-oriented: it is based on worked examples in which the monitoring problem is inherently semantic, contextual, multi-source, and decision-oriented, and in which the ontology layer, analytical services, dashboard interaction, and adaptive logic must work together as a single platform.

A. Worked examples across thematic domains.

IS for public information dissemination. This platform distributes regulatory updates, emergency alerts, and domain-specific messages across web portals, mobile apps, social channels, and APIs. Conventional tools can monitor delivery and latency, but they cannot determine whether a semantically critical message was propagated consistently across all target channels and language versions. The proposed platform supports ontology-driven monitoring by linking message types, audiences, urgency classes, and delivery states.

Financial transaction IS. A financial transaction system processes payments, fraud alerts, compliance rules, account states, and transaction chains. Conventional monitoring can track throughput and latency, but it cannot semantically distinguish technical anomalies from compliance deviations, fraud patterns, or business-process exceptions. The proposed platform supports this distinction by combining telemetry with financial ontology concepts such as transaction type, participant role, compliance state, and risk profile.

Smart building and smart workspace monitoring. A smart workspace integrates environmental sensors, occupancy, equipment status, comfort indicators, and organizational policies. Traditional dashboards can show sensor values and alarms, but they cannot reliably interpret whether a signal combination reflects a critical state or an acceptable contextual variation. The proposed platform supports ontology-driven contextualization by linking physical signals to room type, occupancy pattern, comfort state, and operating mode.

A specification-driven software development environment. A specification-driven development environment includes requirements, formal constraints, test cases, model transformations, deployment logs, and quality metrics. Standard project dashboards track repository activity and CI/CD status, but they do not monitor semantic consistency between requirements, implementation, evidence, and runtime behavior. The proposed platform supports lifecycle-wide IM by treating requirements, traceability links, violations, and software states as semantically integrated monitoring objects.

Multi-agent distributed IS. A multi-agent system contains autonomous agents, roles, goals, negotiation protocols, resource constraints, and changing coordination patterns. Conventional tools can monitor infrastructure and message traffic, but they are weak in detecting whether coordination remains semantically coherent or whether local adaptations cause global strategy drift. The proposed platform supports this monitoring task through semantic representation of incidents, urgency states, actors, resources, and coordination policies.

Industrial cyber-physical manufacturing environment. A production monitoring system integrates PLC data, machine states, maintenance records, production plans, worker context, quality signals, and safety constraints. Existing industrial tools provide telemetry and SCADA-level observability, but often fail to connect anomalies with process goals, operator role, quality risk, and maintenance logic. The proposed platform supports cross-layer monitoring by semantically linking telemetry, production states, constraints, and dashboard recommendations.

Adaptive emergency response, early warning, and crisis coordination platform. A crisis-response IS combines early warning signals, alerts, geospatial data, public communication, responder coordination, and evolving operational context. Traditional tools can monitor technical availability and message delivery, but they cannot determine whether an early warning was classified correctly, triggered the proper response chain, and remained consistent across institutions. The proposed platform supports ontology-driven and adaptive monitoring through semantic representation of incidents, urgency states, actors, resources, and coordination policies.

B. Validation relevance of the proposed examples.

These worked examples validate the proposed platform not only because they belong to different domains, but also because they share common structural properties that justify the proposed architecture. In all these cases, monitoring cannot be reduced to just collecting metrics. Instead, meaningful monitoring requires:

- semantic identification of monitored entities and states;

- formal representation of dependencies, risks, and events;
- combination of multiple analytical modes;
- interpretation and explanation at the dashboard layer;
- adaptive or hierarchical reconfiguration of monitoring logic.

Thus, the examples serve as a validation that the proposed ontology-driven cloud platform is suitable for classes of monitoring problems that are poorly supported by traditional monitoring stacks in their standard form.

C. Validation perspective for future empirical work. For future empirical validation, each of the above examples can be transformed into a scenario benchmark. Such validation can compare the proposed platform with conventional observability environments on criteria such as semantic interpretability, quality of state classification, anomaly contextualization, dashboard-layer decision support, adaptation latency, and completeness of cross-source integration. At the present stage, these criteria define the prototype-level feasibility of the implemented architecture rather than its full empirical validation on real-world projects. Thus, the proposed examples provide not only illustrative use cases but also a structured framework for the next stage of architectural and experimental validation of the proposed IM platform.

X. USER INTERFACE PROTOTYPE OF THE PROPOSED PLATFORM

The platform is currently at the prototype stage, and the present study reports its architecture, core interaction model, and prototype dashboard environment, while full implementation and empirical validation are left for subsequent work.

To illustrate the practical interaction model of the proposed ontology-driven cloud platform for IM, a prototype frontend dashboard with several role-oriented tabs reflecting the main architectural layers and analytical functions of the system was developed. The interface is not limited to conventional monitoring widgets, but is structured as a dashboard-centric analytical environment in which semantic interpretation, multi-method analytics, validation scenarios, and federated monitoring management are presented in a single visual workspace.

The main interface overview (Fig. 6) presents the integrated platform management space.

FIG. 6. Main overview of the dashboard prototype.

It combines global status indicators, layer metrics, semantic alerts, and architectural status dashboards into a single operational view. This screen demonstrates how the platform can support high-level monitoring of the current state of the system while maintaining access to semantically enriched and analytically interpreted information.

The ontology layer screen (Fig. 7) visualizes the semantic integration logic of the platform. It shows how telemetry, context, domain knowledge, ontology resources, rule engines, and dashboard bindings can be represented as related components of a single semantic monitoring pipeline. This interface representation is particularly important because it demonstrates the role of ontologies not as static background models but as operational elements of IM.

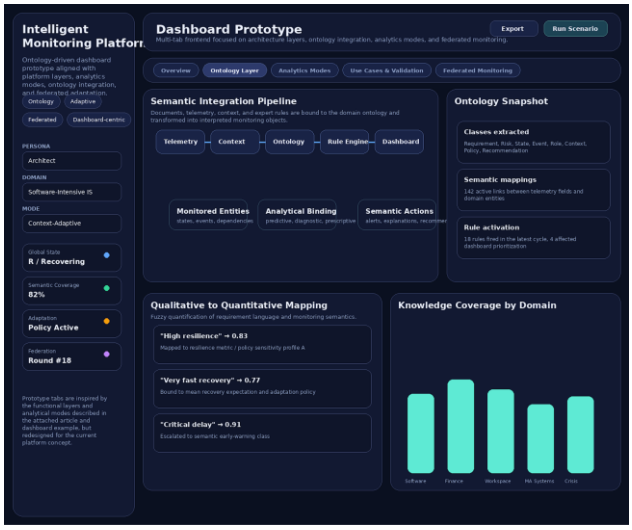

FIG. 7. Ontology layer view of the dashboard prototype.

The analytics modes screen (Fig. 8) reflects the multi-method analytics model proposed in the paper. It integrates descriptive, diagnostic, predictive, prescriptive, comparative, and contextually adaptive analytics into a single dashboard structure.

Thus, the interface supports not only state observation, but also explanation, prediction, response recommendations, and scenario-based reasoning. The validation and use case screen (Fig. 9) presents worked

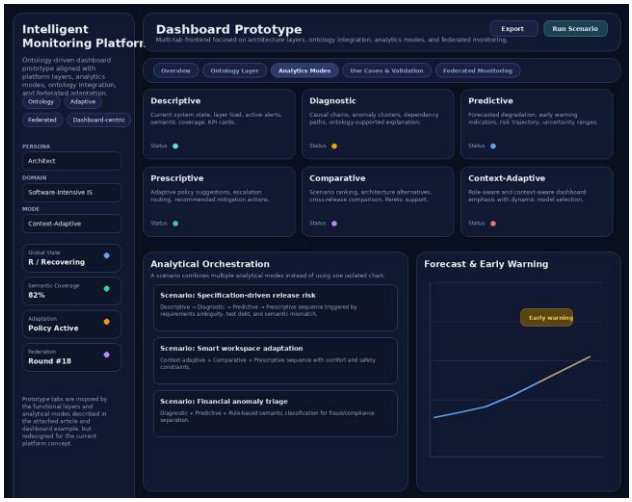

FIG. 8. Analytics modes view of the dashboard prototype.

examples from different subject areas. Its role is to demonstrate that the proposed platform is designed for monitoring tasks where traditional metric-oriented solutions are insufficient and semantic integration, contextual interpretation and adaptive coordination are important. Thus, the interface itself becomes part of the validation of the proposed platform architecture.

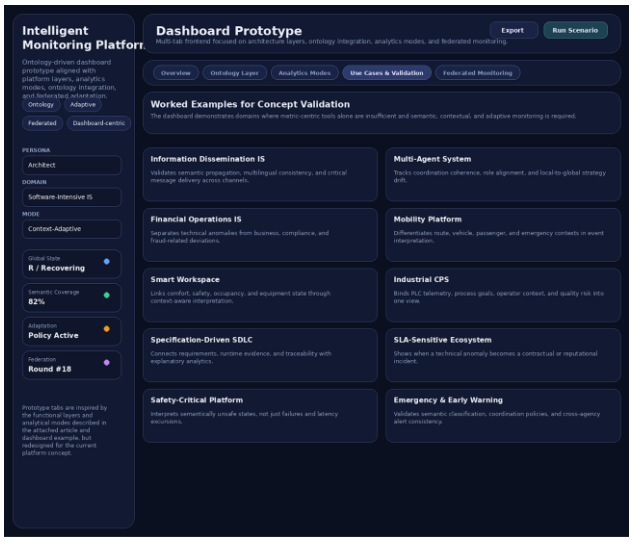

FIG. 9. Validation and use-case view of the dashboard prototype.

Finally, the federated monitoring screen (Fig. 10) illustrates the hierarchical and self-adaptive extension of the platform. It shows how local monitoring nodes, intermediate coordinators, and a global adaptation layer can be represented in the form of a dashboard suitable for distributed IM. This approach supports the interpretation of the proposed platform not only as a centralized monitoring environment, but also as a framework for adaptive federated monitoring management.

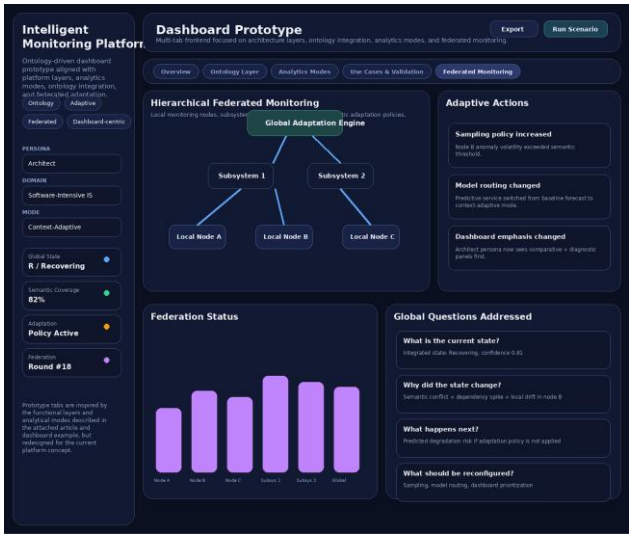

FIG. 10. Federated monitoring view of the dashboard prototype.

Taken together, the proposed interface prototype demonstrates that the platform can be presented as a holistic multi-tabbed analytical environment that follows the architecture described in the paper. The interface supports the transition from isolated metric monitoring to semantically integrated, state-oriented, dashboard-centric, and adaptively coordinated IM.

XI. DISCUSSION

The proposed platform addresses several conceptual and architectural challenges simultaneously. It integrates telemetry, semantics, analytics, and visualization into a single platform model. It also offers a reusable mechanism for unifying subject domains using ontologies instead of reworking the platform for each application area. From a scientific perspective, the most important shift is the transition from fragmented, metrics-driven monitoring to intelligent, domain-driven, semantically integrated, and dashboard-driven monitoring. This shift allows monitoring to support richer interpretation, explanatory power, and adaptive decision-making.

The platform also paves the way for a new class of monitoring environments, in which dashboards serve as knowledge-driven management interfaces, ontologies act as operational integration resources, and monitoring processes themselves become adaptive and hierarchical.

Compared to modern cloud-based monitoring and observability platforms (Table 1), the proposed architecture operates at a different functional level. Modern solutions (OpenTelemetry-based toolchains, Prometheus with Grafana, Amazon CloudWatch, Azure Monitor, Elastic Observability, New Relic, and Dynatrace) provide mature support for telemetry data collection, dashboards, alerting, and, in some cases, AI-driven operational analysis [33-40]. However, these platforms remain focused primarily on platform observability and telemetry workflows. In contrast, the proposed platform is designed as a semantically integrated monitoring environment in which telemetry is explicitly linked to domain entities, monitored states, risks, rules, explanations, and adaptive monitoring policies.

A particularly important distinction is that the proposed platform is designed not only for observing the object of monitoring but also for adapting the monitoring process itself when the monitored object drifts, evolves, or changes its structural, behavioral, or semantic state. In this sense, the platform supports not only observability-oriented monitoring but also more advanced forms of intelligent monitoring in evolving information systems, including the adaptive synthesis of monitoring models for LLM-modified systems, multi-drift predictive monitoring, and the evolution-aware specification-driven synthesis of intelligent monitoring systems [41-43].

Therefore, the proposed approach should be interpreted not as a replacement for standard observability stacks, but as a higher-level ontology-driven platform for monitoring scenarios requiring semantic integration, explicit state-aware interpretation, dashboard-centric analytical reasoning, and adaptive reconfiguration of monitoring logic. Accordingly, the proposed platform is especially relevant for monitoring tasks in which operational signals alone are insufficient and must be interpreted jointly with domain rules, evolving states, and adaptive monitoring policies.

Thus, a comparative analysis demonstrates that the proposed platform differs from modern monitoring tools not by replacing their observability functions, but by extending them toward semantically integrated, state-aware, evolution-aware, and self-reconfiguring intelligent

TABLE 1. Comparative analysis of the proposed platform with contemporary cloud-based monitoring and observability approaches.

Platform / approach	Strengths / advantages	Main limitation relative to the proposed platform
Prometheus + Grafana [33, 34]	Mature open-source ecosystem for metrics collection, visualization, and alerting; flexible integration with cloud-native infrastructures	Does not provide ontology-driven semantic integration, explicit state-aware interpretation, or reconfiguration of monitoring logic when the monitored object drifts or evolves
OpenTelemetry-based toolchains [35]	Standardized and vendor-neutral telemetry instrumentation across heterogeneous systems; suitable as a common observability backbone	Acts mainly as telemetry infrastructure and does not adapt monitoring models or policies to the evolving state of the monitored object
Amazon CloudWatch [36]	Deep integration with AWS services; native dashboards, alarms, events, and large-scale cloud monitoring support	Focuses on cloud-service and infrastructure observability rather than ontology-driven domain reasoning and adaptive reconfiguration of monitoring under object evolution
Azure Monitor [37]	Unified telemetry collection and operational monitoring for Azure and hybrid environments; strong dashboard and alerting support	Does not include ontology-driven semantic integration or monitoring-model adaptation when the monitored system changes structurally or contextually
Elastic Observability [38]	Powerful cross-signal observability with logs, metrics, traces, and search-centric analysis; suitable for operational investigation	Semantics are schema-based rather than ontology-driven, and the platform does not focus on adaptive state-aware monitoring of evolving objects
New Relic [39]	Mature observability interface for metrics, events, logs, and traces, with integrated operational intelligence features	Domain semantics remain implicit in telemetry entities, and monitoring logic is not designed to be re-synthesized as the monitored object evolves
Dynatrace [40]	Advanced causation analysis, AI-assisted diagnostics, and deep full-stack observability across enterprise environments	Provides dependency-aware operational intelligence, but not a reusable ontology-driven architecture that explicitly reconfigures monitoring according to monitored-object drift or evolution

monitoring. Its key advantage is the ability to adapt monitoring models, rules, and coordination logic as the monitored object itself drifts, evolves, or changes its state in a manner relevant to the domain. This makes the proposed platform suitable not only for observability-oriented monitoring, but also for evolution-aware, drift-sensitive, and synthesis-oriented monitoring of changing information systems [41-43].

A limitation of this study is that although the platform has been implemented at the prototype level, it has not yet been validated on a real industrial project. Therefore, the main remaining limitation concerns empirical validation in operational settings rather than the absence of implementation. The paper presents an implemented platform prototype, its semantic integration logic, dashboard-centric analytical interpretation, and hierarchical federated extension, but it does not yet provide a full-scale empirical validation on industrial datasets or a comparative benchmark with existing monitoring platforms. Furthermore, the practical effectiveness of the approach will depend on the quality of ontology engineering, the correctness of semantic mappings, the availability of representative telemetry, and the cost of setting up domain-specific analytics services. These limitations do not diminish the value of the proposed framework, but indicate that its current contribution should be interpreted as an implemented prototype and reference architecture rather than as a fully validated solution ready for industrial deployment.

However, several implementation challenges remain. These include ontology development efforts, the quality of semantic mappings, the management of evolving domain knowledge, inter-service synchronization, model drift, and the evaluation of adaptation policies. These issues should be addressed in future empirical studies.

Furthermore, the proposed platform was validated through a series of related experimental studies. Specifically, its architectural principles were applied in experiments involving the adaptive synthesis of intelligent monitoring models for LLM-modified information systems – addressing architectural changes and monitoring property drift, including a controlled Oracle-to-PostgreSQL migration scenario – as well as in experiments on the evolutionary synthesis of specification-based intelligent monitoring systems using large language models, where monitoring specifications, configurations, and artifacts were generated and regenerated in response to system evolution. These results indicate that the proposed platform is not only theoretically sound but also practically applicable as a prototype-level implementation serving as a foundation for advanced, evolution-oriented, and synthesis-oriented intelligent monitoring workflows [41, 43].

XII. CONCLUSIONS AND FUTURE WORK

This paper presents an implemented prototype and architectural framework for an ontology-driven cloud platform for IM with dashboard-centric analytics and domain knowledge integration. The platform has already been implemented at the prototype level, although it has

not yet been validated on a real industrial project. The proposed platform was designed not as a conventional telemetry collection environment, but as a knowledge-based monitoring space in which heterogeneous observations, semantic resources, analytical services, and adaptive management mechanisms are unified within a single cloud architecture.

The study argues that effective IM in modern ISs requires more than metrics, logs, traces, and alerts considered in isolation. It requires a platform that can combine telemetry with domain semantics, interpret monitoring states in context, support multiple analytical modes, and present results through a dashboard that functions as an interface to support operational decisions rather than a passive reporting surface. In this sense, the main value of the proposed approach lies in the transition from fragmented telemetry-based monitoring to semantically integrated, state-oriented, and architecture-oriented monitoring.

The practical value of the proposed approach lies in the reusability of its platform core across various thematic areas without reworking the platform core. Moreover, the architecture is broad enough to support evolution-aware, drift-sensitive, and synthesis-oriented monitoring of changing information systems, including LLM-modified and specification-driven environments [41, 43]. The architecture can serve as a methodological basis for implementing monitoring platforms for software-intensive systems, cyber-physical environments, enterprise services, and other complex ISs.

Future research should focus on the following areas:

- empirical validation on real industrial and organizational datasets;
- formal ontology engineering and semantic quality assessment;
- assessment of dashboard usability and decision support effectiveness;
- implementation and benchmarking of federated monitoring strategies;
- model drift detection and adaptive retraining pipelines;
- integration with digital twins, modeling services, and knowledge-based optimization.

Overall, the proposed study establishes a coherent foundation for the further development of IM platforms, in which cloud infrastructure, ontological knowledge, dashboard-centric analytics, and adaptive monitoring management are considered as interdependent components of a single platform-oriented paradigm.

AUTHOR CONTRIBUTIONS

V.L. – conceptualization, methodology, investigation, writing (original draft preparation), writing (review and editing).

COMPETING INTERESTS

The author declares no competing interests.

REFERENCES

[1] M. Laiq, N. B. Ali, J. Börstler, and E. Engström, "Software analytics for software engineering: A tertiary review," *arXiv*, no. 2410.05796, 2024, doi: 10.48550/arXiv.2410.05796.

[2] J. Caldeira, F. Brito e Abreu, J. Cardoso, R. Simões, T. Oliveira, and J. Reis, "Software development analytics in practice: A systematic literature review," *Archives of Computational Methods in Engineering*, vol. 30, pp. 2041–2080, 2023, doi: 10.1007/s11831-022-09864-y.

[3] M. Y. Lyashkevych, V. Y. Lyashkevych, and R. Y. Shuvar, "Definition and formalization of the software functional state concept throughout the development life cycle," *Electronics and Information Technologies*, no. 32, pp. 151–170, 2025, doi: 10.30970/eli.32.11.

[4] J. Kosińska and K. Zieliński, "Autonomic management framework for cloud-native applications," *Journal of Grid Computing*, vol. 18, pp. 779–796, 2020, doi: 10.1007/s10723-020-09532-0.

[5] J. Lin, Z. Xie, Y. Guo, and J. Xu, "A multi-dimensional extensible cloud-native service stack for enterprises," *Journal of Cloud Computing*, vol. 11, p. 71, 2022, doi: 10.1186/s13677-022-00366-7.

[6] Z. A. Khan, D. Shin, D. Bianculli, et al., "Impact of log parsing on deep learning-based anomaly detection," *Empirical Software Engineering*, vol. 29, p. 139, 2024, doi: 10.1007/s10664-024-10533-w.

[7] G. Kånåhols, S. Hasan, and P. E. Strandberg, "Integrating time series anomaly detection into DevOps workflows," *IEEE Access*, vol. 13, pp. 46459–46477, 2025, doi: 10.1109/ACCESS.2025.3550665.

[8] I. Y. Rohatskyi and V. Y. Lyashkevych, "Semantic integration of models and contexts based on ontology and reinforcement learning in multi-agent distributed computer vision systems," *Information Technology: Computer Science, Software Engineering and Cyber Security*, no. 4, pp. 232–241, 2025, doi: 10.32782/IT/2025-4-27.

[9] M. I. Yakubovych and V. Y. Lyashkevych, "Context obtaining method in sustainable workplaces," *Informatics and Mathematical Methods in Simulation*, vol. 16, no. 1, pp. 34–42, 2026, doi: 10.15276/imms.v16.no1.34.

[10] V. Lyashkevych and R. Shuvar, "Exploratory data analysis possibility in the meaning space using large language models," *Electronics and Information Technologies*, vol. 25, no. 9, pp. 102–116, 2024, doi: 10.30970/eli.25.9.

[11] E. Aguado, et al., "A survey of ontology-enabled processes for dependable robot autonomy," *Computational Intelligence in Robotics*, vol. 11, p. 1377897, 2024, doi: 10.3389/frobt.2024.1377897.

[12] E. Atencio, et al., "Ontology-Based Integration of Enterprise Architecture and Project Management. Systems," *Systems*, vol. 13, no. 6, p. 477, 2025, doi: 10.3390/systems13060477.

[13] B. Di Martino, et al., "CloudOntoViz: A Browser for Knowledge-Based Representations of Cloud Patterns and Services," *IJSWIS*, vol. 21, no. 1, pp. 1–24, 2025, doi: 10.4018/IJSWIS.392474.

[14] S. Loshkovska and P. Panov, "Foundations for a generic ontology for visualization: A comprehensive survey," *Information*, vol. 16, no. 10, p. 915, 2025, doi: 10.3390/info16100915.

[15] L. Ben-Shimol, et al., "Observability and incident response in managed serverless environments using ontology-based log monitoring," *IEEE Transactions on Cloud Computing*, vol. 13, no. 4, pp. 1161–1176, 2025, doi: 10.1109/TCC.2025.3598060.

[16] S. Banerjee, C. E. Fullerton, S. S. Gaharwar, and E. J. Jaselskis, "Strategic web-based data dashboards as monitoring tools for promoting organizational innovation," *Buildings*, vol. 15, no. 13, p. 2204, 2025, doi: 10.3390/buildings15132204.

[17] O. Johnphill, et al., “Cross: A cloud-native approach to automated remediation and observability,” *Journal of Cloud Computing*, 2026, doi: 10.1186/s42400-026-00549-8.
[18] L. Li, L. Zhu, and W. Li, “HiSatFL: A hierarchical federated learning framework for satellite networks with cross-domain privacy adaptation,” *Electronics*, vol. 14, no. 16, p. 3237, 2025, doi: 10.3390/electronics14163237.
[19] A. Aljohani, O. F. Rana, and C. Perera, “Self-adaptive federated learning in Internet of Things systems: A review,” *ACM Computing Surveys*, vol. 57, no. 10, p. 252, 2025, doi: 10.1145/3725527.
[20] S. S. Shahid, T. Salman, M. Baza, and M. A. Teixeira, “Hierarchical federated learning: Approaches, applications, and open challenges,” *IEEE Access*, vol. 13, pp. 190333–190353, 2025, doi: 10.1109/ACCESS.2025.3628455.
[21] V. Gura, O. Ostrovska, and V. Lyashkevych, “Scenario modelling and impact estimation of a local pollutant on the environment,” *Applied Computer Systems*, vol. 30, no. 1, pp. 195–201, 2025, doi: 10.2478/acss-2025-0021.
[22] S. N. Stoyanov and B. L. Belichev, “Building a regional platform for monitoring air quality,” *Future Internet*, vol. 18, no. 2, p. 78, 2026, doi: 10.3390/fi18020078.
[23] M. Segura-Cedres, et al., “An ontology proposal for implementing digital twins in hospitality systems,” *Sensors*, vol. 25, no. 14, p. 4504, 2025, doi: 10.3390/s25144504.
[24] M. Segura-Cedres, D. Manzano-Farray, C. L. Aguiar-Castillo, R. Perez-Jimenez, V. M. Icaza, E. Niarchou, and V. Guerra-Yanez, “An ontology-driven digital twin for hotel front desk: Real-time integration of wearables and OCC camera events via a property-defined REST API,” *Electronics*, vol. 15, no. 3, p. 567, 2026, doi: 10.3390/electronics15030567.
[25] J. Schlenger, K. Pluta, A. Mathew, T. Yeung, R. Sacks, and A. Borrmann, “Reference architecture and ontology framework for digital twin construction,” *Automation in Construction*, vol. 174, p. 106111, 2025, doi: 10.1016/j.autcon.2025.106111.
[26] S. V. Holub and D. V. Tolbatov, “Metod syntezu bahatosharovoi modeli monitorynhovoho prohramnoho ahenta,” *Matematychni Mashyny i Systemy*, no. 1, pp. 101–111, 2023, doi: 10.34121/1028-9763-2023-1-101-111.
[27] S. V. Holub and D. V. Tolbatov, “Udoskonalennia metodu syntezu bahatosharovykh modelei monitorynhovoho prohramnoho ahenta,” *Aktualni Problemy Avtomatyzatsii ta Informatsiinykh Tekhnolohii*, no. 27, pp. 53–60, 2023, doi: 10.15421/432306.
[28] S. V. Holub and V. V. Ostapiuk, “Mashynne navchannia bahatosharovykh modelei monitorynhovoho prohramnoho ahenta,” *Matematychni Mashyny i Systemy*, no. 2, pp. 76–95, 2025, doi: 10.34121/1028-9763-2025-2-76-95.
[29] S. Kunytska and S. Holub, “Multi-agent monitoring information systems,” in *Mathematical Modeling and Simulation of Systems (MODS 2019)*, vol. 1019, Advances in Intelligent Systems and Computing, pp. 164–171, 2020, doi: 10.1007/978-3-030-25741-5_17.
[30] M. V. Talakh, S. V. Holub, P. O. Luchshev, and I. B. Turkin, “Intelligent monitoring of air temperature by the data of satellites and meteorological stations,” *International Journal of Computing*, vol. 21, no. 1, pp. 120–127, 2022, doi: 10.47839/ijc.21.1.2525.
[31] Y. V. Rogushina, “The use of ontological knowledge for multi-criteria comparison of complex information objects,” *Problems in Programming*, no. 2, pp. 82–95, 2017, doi: 10.15407/pp2022.03-04.249.
[32] O. V. Nesterenko, “Ontology-driven information systems in administrative management,” *Mathematical Modeling in Economics*, no. 2, no. 15, pp. 57–68, 2019.
[33] Prometheus Authors, “Overview | Prometheus,” *Prometheus Documentation*. [Online]. Available: https://prometheus.io/docs/introduction/overview/.
[34] Grafana Labs, “About Grafana,” *Grafana Documentation*. [Online]. Available: https://grafana.com/docs/grafana/latest/introduction/.
[35] OpenTelemetry Authors, “Documentation | OpenTelemetry,” *OpenTelemetry*. [Online]. Available: https://opentelemetry.io/docs/.
[36] Amazon Web Services, “What is Amazon CloudWatch?,” *Amazon CloudWatch Documentation*. [Online]. Available: https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html.
[37] Microsoft, “Azure Monitor overview,” *Microsoft Learn*. [Online]. Available: https://learn.microsoft.com/en-us/azure/azure-monitor/fundamentals/overview.
[38] Elastic, “Elastic Observability solution overview,” *Elastic Docs*. [Online]. Available: https://www.elastic.co/docs/solutions/observability.
[39] New Relic, “New Relic data types: metrics, events, logs, and traces (MELT),” *New Relic Documentation*. [Online]. Available: https://docs.newrelic.com/docs/data-apis/understand-data/new-relic-data-types/. Accessed: Jul. 3, 2026.
[40] Dynatrace, “Root cause analysis,” *Dynatrace Documentation*. [Online]. Available: https://docs.dynatrace.com/docs/dynatrace-intelligence/root-cause-analysis.
[41] V. Lyashkevych, “Adaptive synthesis of intelligent monitoring models for LLM-modified information systems: A genetic optimisation approach,” *Applied Computer Systems*, vol. 31, no. 1, pp. 116–126, 2026, doi: 10.2478/acss-2026-0010.
[42] V. Lyashkevych, “Multi-drift predictive monitoring for evolving information systems,” *Computer Systems and Information Technologies*, no. 2, pp. 171–184, 2026, doi: 10.31891/csit-2026-2-15.
[43] V. Lyashkevych, “Evolution-aware specification-driven synthesis of intelligent monitoring systems by large language models,” *Informatics and Mathematical Methods in Simulation*, vol. 16, no. 3, pp. 424–432, 2026, doi: 10.15276/imms.v16.no3.424.

Vasyl Lyashkevych

Received a Master's degree from Chernivtsi National University in 2000. Defended PhD thesis in Computer Science in 2007. Since 2021, works as an Associate Professor at Ivan Franko National University of Lviv. Research interests include artificial intelligence, intelligent monitoring, multi-agent systems, ontological modeling, large language models, and software engineering technologies.

ORCID ID: 0000-0003-2810-6061

Керована онтологією хмарна платформа для інтелектуального моніторингу з аналітикою, орієнтованою на інформаційну панель, та інтеграцією знань предметної області

Василь Ляшкевич*

Кафедра системного проектування, факультет електроніки та комп'ютерних технологій,
Львівський національний університет імені Івана Франка, Львів, Україна

*Автор-кореспондент (Електронна адреса: vasyl.liashkevych@lnu.edu.ua)

АНОТАЦІЯ У цій статті представлено хмарну платформу на основі онтологій для інтелектуального моніторингу, яка об'єднує представлення знань предметної області, хмарний телеметричний збір, багатометодульну аналітику та підтримку рішень на основі інформаційних панелей в єдиному адаптивному середовищі. На відміну від традиційних рішень для моніторингу, які в основному агрегують метрики, журнали та сповіщення, запропонована платформа розглядає контрольовану інформаційну систему як динамічний, залежний від предметної області об'єкт, стан якого необхідно спостерігати, діагностувати, прогнозувати та семантично інтерпретувати з урахуванням контексту, архітектури та операційних обмежень. Платформа інтегрує онтології предметної області, телеметричні конвеєри, аналітичні та моделюючі служби, ієрархічні вузли моніторингу та інтерактивний шар інформаційної панелі, що підтримує описову, діагностичну, прогнозну, прескриптивну, порівняльну та контекстно-адаптивну аналітику. Дослідження мотивоване переходом від фрагментованого моніторингу, орієнтованого на метрики, до моніторингу на основі знань, здатного поєднувати гетерогенні джерела даних, семантику предметної області та адаптивну підтримку рішень. Архітектура організована як багаторівнева хмарна платформа, що включає збір даних, інтеграцію та потокову передачу, зберігання, управління онтологіями та знаннями, аналітичні служби, ієрархічний федеративний моніторинг, самоадаптацію та взаємодію з інформаційними панелями. Центральною особливістю платформи є інтеграція знань предметної області в цикл моніторингу за допомогою онтологічних зіставлень, семантичного збагачення телеметрії, інтерпретації подій на основі правил та віджетів панелі інструментів, що ґрунтуються на концепціях предметної області. Це дозволяє платформі не лише виявляти аномалії та моделі деградації, але й пояснювати їх мовою цільової області та адаптувати конфігурацію моніторингу до змін у стані контрольованої системи. У статті також окреслено тематичні варіанти використання, що показують, як одне й те саме ядро платформи може бути підключене до різних доменних областей за допомогою онтологічних ресурсів та специфічних для предметної області аналітичних шаблонів. Запропонований підхід забезпечує довідкову основу для семантично інтегрованих платформ моніторингу.

КЛЮЧОВІ СЛОВА інтелектуальний моніторинг, аналітична панель інструментів, інтеграція знань предметної області, федеративний моніторинг, самоадаптивний моніторинг.